```
                              H  5



    FFFFFFFFFFFFFFF      111         111    XXX         XXX
    FFFFFFFFFFFFFFF      111         111    XXX         XXX
    FFFFFFFFFFFFFFF      111         111    XXX         XXX
    FFF               111111      111111    XXX         XXX
    FFF               111111      111111    XXX         XXX
    FFF               111111      111111    XXX         XXX
    FFF                  111         111       XXX   XXX
    FFF                  111         111       XXX   XXX
    FFF                  111         111       XXX   XXX
    FFFFFFFFFFFF         111         111          XXX
    FFFFFFFFFFFF         111         111          XXX
    FFFFFFFFFFFF         111         111          XXX
    FFF                  111         111       XXX   XXX
    FFF                  111         111       XXX   XXX
    FFF                  111         111       XXX   XXX
    FFF                  111         111    XXX         XXX
    FFF                  111         111    XXX         XXX
    FFF                  111         111    XXX         XXX
    FFF               111111111   111111111 XXX         XXX
    FFF               111111111   111111111 XXX         XXX
    FFF               111111111   111111111 XXX         XXX
```

```
_$25

Symb
----
IOC1
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

```
**FILE**ID**DELFIL                                        M 9




DDDDDDDD    EEEEEEEEEE  LL          FFFFFFFFFF    IIIIII    LL
DDDDDDDD    EEEEEEEEEE  LL          FFFFFFFFFF    IIIIII    LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EEEEEEEE    LL          FFFFFFFF        II      LL
DD      DD  EEEEEEEE    LL          FFFFFFFF        II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL               ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  FF            IIIIII    LLLLLLLLLL       ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  FF            IIIIII    LLLLLLLLLL       ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

DE
VO

```
0 MODULE DELFIL (
0               LANGUAGE (BLISS32),
0               IDENT = 'V04-000'
0               ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This module deletes a file, returning its blocks to the storage map
1 !       and releasing the file header.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  4-Apr-1977  15:50
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-012 CDS0008         Christian D. Saether    22-Aug-1984
1 !               Don't complain about directories either (CDS0006).
1 !
1 !       V03-011 ACG0444         Andrew C. Goldstein,    21-Aug-1984  20:43
1 !               Fix error recovery in file ID cache flush code
1 !
```

```
1 !     V03-010 CDS0007         Christian D. Saether    14-Aug-1984
1 !             Don't complain (CDS0006) about extension headers.
1 !
1 !     V03-009 CDS0006         Christian D. Saether    10-Aug-1094
1 !             Add bugchecks to guard against deleting the wrong file,
1 !             and directories in particular.
1 !
1 !     V03-008 CDS0005         Christian D. Saether    7-Aug-1984
1 !             Replace TOSS_CACHE_DATA call with KILL_BUFFERS call.
1 !
1 !     V03-007 ACG0438         Andrew C. Goldstein,    1-Aug-1984  17:14
1 !             Add cache interlock logic
1 !
1 !     V03-006 ACG0409         Andrew C. Goldstein,    22-Mar-1984  0:08
1 !             Don't invalidate deleted file headers, as they are
1 !             likely to be reused soon, due to the file ID cache.
1 !             Make APPLY_RVN and DEFAULT_RVN macros.
1 !
1 !     V03-005 CDS0004         Christian D. Saether     1-Mar-1984
1 !             Replace call to FLUSH_FID with call to TOSS_CACHE_DATA.
1 !
1 !     V03-004 CDS0003         Christian D. Saether    29-Dec-1983
1 !             Use L_NORM linkage and BIND_COMMON macro.
1 !
1 !     V03-003 CDS0002         Christian D. Saether    13-Sep-1983
1 !             Change interface to allocation serialization.
1 !
1 !     V03-002 CDS0001         Christian D. Saether    13-May-1983
1 !             Serialize file header deletion processing.
1 !
1 !     V03-001 LMP0077         L. Mark Pilant,         31-Jan-1983  10:26
1 !             Eliminate the check made for extension headers as this is
1 !             now done in the DELETE module.  An access conflict error
1 !             will result if an attempt is made to delete a file that
1 !             has one of its extension headers accessed.
1 !
1 !     V02-007 ACG0229         Andrew C. Goldstein,    23-Dec-1981  21:59
1 !             Count file ID cache hits and misses
1 !
1 !     V02-006 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:25
1 !             Previous revision history moved to F11B.REV
1 !**
1
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 REQUIRE 'SRC$:FCPDEF.B32';
1
1
1 FORWARD ROUTINE
1       DELETE_FILE     : L_NORM NOVALUE, ! complete file deletion
1       DELETE_FID      : L_NORM NOVALUE, ! just release file header
1       RETURN_FILE_NUM : L_NORM,         ! return file number to cache
1       REMOVE_FILE_NUM : L_NORM;         ! remove file numbers from cache
```

```
1 GLOBAL ROUTINE DELETE_FILE (FIB, FILEHEADER) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine deletes a file by releasing its blocks to the storage
1 !       bitmap and then releasing the header.
1 !
1 ! CALLING SEQUENCE:
1 !       DELETE_FILE (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: FIB of operation
1 !       ARG2: address of file header buffer
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       File deleted, storage map and index file bitmap modified, VCB modified
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FIB             : REF BBLOCK,   ! address of user FIB
2       FILEHEADER      : REF BBLOCK;   ! address of file header
2
2 LOCAL
2       HEADER          : REF BBLOCK,   ! local address of file header
2       FCB             : REF BBLOCK,   ! FCB of header in process
2       FILE_NUMBER,                    ! file number of header being deleted
2       EXT_FID         : BBLOCK [FID$C_LENGTH], ! extension file ID
2       EX_SEGNUM,                      ! header extension segment number
2       FILESIZE;                       ! size of file section
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2       FILE_SIZE       : L_NORM,       ! compute file section size
2       CHARGE_QUOTA    : L_NORM,       ! charge user's disk quota
2       CHECKSUM        : L_NORM,       ! compute file header checksum
2       SEND_BADSCAN    : L_NORM ADDRESSING_MODE (GENERAL),
2                                       ! start bad block scan process
2       WRITE_BLOCK     : L_NORM,       ! write block to disk
2       TRUNCATE_HEADER : L_NORM,       ! truncate file header
```

```
2               NEXT_HEADER     : L_NORM;               ! read next file extension header
2
2
2       HEADER = .FILEHEADER;
2
3       IF (.HEADER [FH2$W_SEG_NUM] EQL 0
3           AND (.(FIB [FIB$W_FID]) NEQ .(HEADER [FH2$W_FID])    ! fid_num + fid_seq
3                OR .FIB [FIB$B_FID_NMX] NEQ .HEADER [FH2$B_FID_NMX]))
3
2       THEN
2           BUG_CHECK (WRTINVBUF, 'attempted to delete the wrong file');
2
2       ! If the file is marked bad and is not empty, we do not delete the file,
2       ! but rather send it to the bad block scanner, who will analyze the file and
2       ! delete it piecemeal.
2       !
2
2       IF   .HEADER[FH2$V_BADBLOCK]
3       AND (    .HEADER[FH2$B_MAP_INUSE] NEQ 0
3           OR   .HEADER[FH2$W_EX_FIDNUM] NEQ 0
3           OR   .HEADER[FH2$W_EX_FIDRVN] NEQ 0)
2       THEN
2           BEGIN
3           CHECKSUM (.HEADER);
3           SEND_BADSCAN (HEADER[FH2$W_FID]);
3           RETURN;
2           END;
2
2       ! Loop for all headers, releasing the blocks mapped and the headers.
2       !
2
2       WHILE 1 DO
3           BEGIN
3           FILE_NUMBER = .HEADER[FH2$W_FID_NUM];
3           IF .CURRENT_VCB[VCB$V_EXTFID]
3           THEN FILE_NUMBER<16,8> = .HEADER[FH2$B_FID_NMX];
3           NEW_FID = .FILE_NUMBER;                 ! record file number for cleanup
3           NEW_FID_RVN = .CURRENT_RVN;
3
3           HEADER[FH2$W_FID_NUM] = 0;              ! deleted header has zero file number
3           HEADER[FH2$W_FID_RVN] = 0;
3           HEADER[FH2$W_CHECKSUM] = 0;             ! and zero checksum
3           FILE_HEADER = 0;
3           WRITE_BLOCK (.HEADER);
3
3       ! Credit the header and the blocks it maps to the owner's disk quota.
3       !
3
3           FILESIZE = 0;
3           IF NOT .CLEANUP_FLAGS[CLF_NOTCHARGED]
3           THEN FILESIZE = FILE_SIZE (.HEADER);
3           IF NOT .CLEANUP_FLAGS[CLF_HDRNOTCHG]
3           THEN FILESIZE = .FILESIZE + 1;
3           CHARGE_QUOTA (.HEADER[FH2$L_FILEOWNER], -.FILESIZE, BITLIST (QUOTA_CHARGE));
3
3       ! Now return the blocks mapped by the header to the storage map.
3       ! Then extract the extension header data.
```

```
;   226          1215  3 !
;   227          1216  3
;   228          1217  3     TRUNCATE_HEADER (.FIB, .HEADER);
;   229          1218  3
;   230          1219  3     EX_SEGNUM = .HEADER[FH2$W_SEG_NUM] + 1;
;   231          1220  3     CH$MOVE (FID$C_LENGTH, HEADER[FH2$W_EXT_FID], EXT_FID);
;   232          1221  3
;   233          1222  3 ! Now free the header in the index file bitmap. Then chain to the next header,
;   234          1223  3 ! if any, and repeat.
;   235          1224  3 !
;   236          1225  3
;   237          1226  3     NEW_FID = 0;
;   238          1227  3     DELETE_FID (.FILE_NUMBER);
;   239          1228  3
;   240          1229  3     HEADER = NEXT_HEADER (0, 0, EXT_FID, .EX_SEGNUM);
;   241          1230  3     IF .HEADER EQL 0 THEN EXITLOOP;
;   242          1231  2     END;
;   243          1232  2
;   244          1233  1 END;                                        ! end of routine DELETE_FILE

                                                                      .TITLE  DELFIL
                                                                      .IDENT  \V04-000\

                                                                      .EXTRN  FILE_SIZE, CHARGE_QUOTA
                                                                      .EXTRN  CHECKSUM, SEND_BADSCAN
                                                                      .EXTRN  WRITE_BLOCK, TRUNCATE_HEADER
                                                                      .EXTRN  NEXT_HEADER, BUG$_WRTINVBUF

                                                                      .PSECT  $CODE$,NOWRT,2

                                          03FC 00000                  .ENTRY  DELETE_FILE, Save R2,R3,R4,R5,R6,R7,R8,R9   ; 1101
                  5E                   08   C2 00002                  SUBL2   #8, SP                                      ;
                  56           08   AC   D0 00005                     MOVL    FILEHEADER, HEADER                          ; 1161
                  04           A6   B5 00009                          TSTW    4(HEADER)                                   ; 1163
                               16   12 0000C                          BNEQ    2$                                          ;
                  50      04   AC   D0 0000E                          MOVL    FIB, R0                                     ; 1164
            08    A6      04   A0   D1 00012                          CMPL    4(R0), 8(HEADER)                            ;
                               07   12 00017                          BNEQ    1$                                          ;
            0D    A6      09   A0   91 00019                          CMPB    9(R0), 13(HEADER)                           ; 1165
                               04   13 0001E                          BEQL    2$                                          ;
                                  FEFF 00020 1$:                      BUGW                                                ; 1168
                                 0000* 00022                          .WORD   <BUG$_WRTINVBUF!4>                          ;
  21        35    A6           06   E1 00024 2$:                      BBC     #6, 53(HEADER), 4$                          ; 1175
                          3A   A6   95 00029                          TSTB    58(HEADER)                                  ; 1176
                               0A   12 0002C                          BNEQ    3$                                          ;
                          0E   A6   B5 0002E                          TSTW    14(HEADER)                                  ; 1177
                               05   12 00031                          BNEQ    3$                                          ;
                          12   A6   B5 00033                          TSTW    18(HEADER)                                  ; 1178
                               12   13 00036                          BEQL    4$                                          ;
                               56   DD 00038 3$:                      PUSHL   HEADER                                      ; 1181
         0000G    CF           01   FB 0003A                          CALLS   #1, CHECKSUM                                ;
                          08   A6   9F 0003F                          PUSHAB  8(HEADER)                                   ; 1182
     00000000G    00           01   FB 00042                          CALLS   #1, SEND_BADSCAN                            ;
                                    04 00049                          RET                                                 ; 1180
                  57      08   A6   3C 0004A 4$:                      MOVZWL  8(HEADER), FILE_NUMBER                      ; 1191
                  50      98   AA   D0 0004E                          MOVL    -104(BASE), R0                              ; 1192
```

```
                         06      0B   A0          05   E1 00052          BBC     #5, 11(R0), 5$                                   ;
          57             08           10    0D    A6   F0 00057          INSV    13(HEADER), #16, #8, FILE_NUMBER                 ; 1193
                                 A8   AA          57   D0 0005D 5$:      MOVL    FILE_NUMBER, -88(BASE)                           ; 1194
                                 AC   AA    A0    AA   D0 00061          MOVL    -96(BASE), -84(BASE)                             ; 1195
                                            08    A6   B4 00066          CLRW    8(HEADER)                                        ; 1197
                                            0C    A6   B4 00069          CLRW    12(HEADER)                                       ; 1198
                                          01FE    C6   B4 0006C          CLRW    510(HEADER)                                      ; 1199
                                            04    AA   D4 00070          CLRL    4(BASE)                                          ; 1200
                                                  56   DD 00073          PUSHL   HEADER                                           ; 1201
                             0000G   CF           01   FB 00075          CALLS   #1, WRITE_BLOCK                                  ;
                                                  58   D4 0007A          CLRL    FILESIZE                                         ; 1206
                         0A           6A          1D   E0 0007C          BBS     #29, (BASE), 6$                                  ; 1207
                                                  56   DD 00080          PUSHL   HEADER                                           ; 1208
                             0000G   CF           01   FB 00082          CALLS   #1, FILE_SIZE                                    ;
                                      58          50   D0 00087          MOVL    R0, FILESIZE                                     ;
                         02           6A          1B   E0 0008A 6$:      BBS     #27, (BASE), 7$                                  ; 1209
                                                  58   D6 0008E          INCL    FILESIZE                                         ; 1210
                                                  02   DD 00090 7$:      PUSHL   #2                                               ; 1211
                                      7E          58   CE 00092          MNEGL   FILESIZE, -(SP)                                  ;
                                            3C    A6   DD 00095          PUSHL   60(HEADER)                                       ;
                             0000G   CF           03   FB 00098          CALLS   #3, CHARGE_QUOTA                                 ;
                                                  56   DD 0009D          PUSHL   HEADER                                           ; 1217
                                            04    AC   DD 0009F          PUSHL   FIB                                              ;
                             0000G   CF           02   FB 000A2          CALLS   #2, TRUNCATE_HEADER                              ;
                                      59    04    A6   3C 000A7          MOVZWL  4(HEADER), EX_SEGNUM                             ; 1219
                                                  59   D6 000AB          INCL    EX_SEGNUM                                        ;
                         6E      0E   A6          06   28 000AD          MOVC3   #6, 14(HEADER), EXT_FID                          ; 1220
                                            A8    AA   D4 000B2          CLRL    -88(BASE)                                        ; 1226
                                                  57   DD 000B5          PUSHL   FILE_NUMBER                                      ; 1227
                             0000V   CF           01   FB 000B7          CALLS   #1, DELETE_FID                                   ;
                                                  59   DD 000BC          PUSHL   EX_SEGNUM                                        ; 1229
                                            04    AE   9F 000BE          PUSHAB  EXT_FID                                          ;
                                                  7E   7C 000C1          CLRQ    -(SP)                                            ;
                             0000G   CF           04   FB 000C3          CALLS   #4, NEXT_HEADER                                  ;
                                      56          50   D0 000C8          MOVL    R0, HEADER                                       ;
                                                  03   13 000CB          BEQL    8$                                               ; 1230
                                              FF7A     31 000CD          BRW     4$                                               ;
                                                       04 000D0 8$:      RET                                                      ; 1233

; Routine Size:  209 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 GLOBAL ROUTINE DELETE_FID (FILENUM) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine marks the indicated file header free in the index
1 !       file bitmap.
1 !
1 ! CALLING SEQUENCE:
1 !       DELETE_HEADER (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: file number of header
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: VCB of volume
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       Header deleted - index file bitmap & VCB altered
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       FP;
2
2 LOCAL
2       CACHE           : REF BBLOCK,   ! address of cache block
2       FID_CACHE       : REF BBLOCK,   ! address of file number cache
2       LOCK_STATUS     : VECTOR [2],   ! lock status block
2       VBN,                            ! relative block in bitmap
2       BEST_VBN,                       ! best block to return bits to
2       COUNT,                          ! number of FID's in current block
2       BEST_COUNT,                     ! number if FID's in best block
2       BLOCK,                          ! block number of current entry
2       BUFFER;                         ! bitmap buffer
2
2 EXTERNAL
2       PMS$GL_FIDHIT   : ADDRESSING_MODE (GENERAL),
2                                       ! count of file ID cache hits
2       PMS$GL_FIDMISS  : ADDRESSING_MODE (GENERAL);
2                                       ! count of file ID cache misses
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
```

```
2               ALLOCATION_LOCK : L_NORM,
2               INIT_FID_CACHE  : L_NORM,       ! initialize file ID cache lock
2               READ_BLOCK      : L_NORM,       ! read a block from the disk
2               WRITE_BLOCK     : L_NORM,       ! write it back
2               ZERO_ON_ERROR;                  ! return zero on error signal (handler)
2
2       ! Serialize against other storage or file header allocation/deallocation
2       ! operations.
2       !
2
2       ALLOCATION_LOCK ();
2
2       ! If this is not a flush call, we delete the file number by returning it
2       ! to the file number cache. If the cache fills up, the kernel mode routine
2       ! returns LBC. We then scan the cache, looking for the largest group of file
2       ! numbers that are all in the same bitmap block (up to half of the cache),
2       ! and then flush those from the cache. If this is a cache flush call or
2       ! the volume is marked for dismount, however, we flush the entire cache.
2       !
2
2       CACHE = .CURRENT_VCB[VCB$L_CACHE];
2       FID_CACHE = .CACHE[VCA$L_FIDCACHE];
2
2       IF .FILENUM NEQ 0
2       THEN
3           BEGIN
3           IF NOT .CACHE[VCA$V_FIDC_VALID]
3           THEN INIT_FID_CACHE (.CACHE);
4           IF KERNEL_CALL (RETURN_FILE_NUM, .FILENUM)
3           THEN
4               BEGIN
4               PMS$GL_FIDHIT = .PMS$GL_FIDHIT + 1;
4               RETURN;
3               END;
2           END;
2
2       IF .FILENUM NEQ 0
2       AND .CACHE[VCA$V_FIDC_VALID]
2       THEN
3           BEGIN
3           PMS$GL_FIDMISS = .PMS$GL_FIDMISS + 1;
3           BEST_COUNT = 0;
3           VBN = -1;
3           INCR J FROM 1 TO .FID_CACHE[VCA$W_FIDCOUNT]
3           DO
4               BEGIN
4               BLOCK = (.VECTOR [FID_CACHE[VCA$L_FIDLIST], .J-1] - 1) / 4096;
4               IF .BLOCK NEQ .VBN
4               THEN
5                   BEGIN
5                   VBN = .BLOCK;
5                   COUNT = 0;
4                   END;
4               COUNT = .COUNT + 1;
4               IF .COUNT GTRU .BEST_COUNT
4               THEN
5                   BEGIN
```

```
  5                 BEST_COUNT = .COUNT;
  5                 BEST_VBN = .VBN;
  4                 END;
  4             IF .BEST_COUNT GEQU .FID_CACHE[VCA$W_FIDCOUNT]/2
  4             THEN EXITLOOP;
  3             END;
  3
  3 ! Read the appropriate block, return the desired number of file numbers to
  3 ! it, and write it back.
  3 !
  3
  3         IF .BEST_VBN GEQU .CURRENT_VCB[VCB$B_IBMAPSIZE]
  3         THEN BUG_CHECK (BADFID, FATAL, 'ACP file number out of range for this volume');
  3
  3         BUFFER = READ_BLOCK (.BEST_VBN + .CURRENT_VCB[VCB$L_IBMAPLBN], 1, INDEX_TYPE);
  3         KERNEL_CALL (REMOVE_FILE_NUM, .BEST_COUNT, .BEST_VBN, .BUFFER);
  3         WRITE_BLOCK (.BUFFER);
  3         END
  3
  3 ! If this is a cache flush, loop for all the blocks represented in the
  3 ! cache, read the block, return the file numbers, and write it. Then
  3 ! mark the cache invalid, and release the cache lock if there is one.
  3 ! This operation is done under a handler to ensure its completion in
  3 ! the face of I/O errors.
  3 !
  2 ELSE
  3     BEGIN
  3     .FP = ZERO_ON_ERROR;
  3     UNTIL .FID_CACHE[VCA$W_FIDCOUNT] EQL 0
  3     DO
  4         BEGIN
  4         VBN = (.FID_CACHE[VCA$L_FIDLIST] - 1) / 4096;
  4         IF .VBN GEQU .CURRENT_VCB[VCB$B_IBMAPSIZE]
  4         THEN BUG_CHECK (BADFID, FATAL, 'ACP file number out of range for this volume');
  4
  4         BUFFER = READ_BLOCK (.VBN + .CURRENT_VCB[VCB$L_IBMAPLBN], 1, INDEX_TYPE);
  4         IF .BUFFER NEQ 0
  4         THEN
  5             BEGIN
  5             KERNEL_CALL (REMOVE_FILE_NUM, 0, .VBN, .BUFFER);
  5             WRITE_BLOCK (.BUFFER);
  5             END
  4         ELSE
  4             FID_CACHE[VCA$W_FIDCOUNT] = 0;
  3         END;
  3     IF .FID_CACHE[VCA$L_FIDCLKID] NEQ 0
  3     THEN
  4         BEGIN
  4         LOCK_STATUS[1] = .FID_CACHE[VCA$L_FIDCLKID];
P 4         IF NOT $ENQW (EFN      = EFN,
P 4                       LKMODE = LCK$K_NLMODE,
P 4                       FLAGS  = LCK$M_NOQUEUE OR LCK$M_SYNCSTS OR LCK$M_CONVERT OR LCK$M_CVTSYS,
P 4                       LKSB   = LOCK_STATUS
  5                       )
  4         THEN BUG_CHECK (XQPERR, FATAL, 'Unexpected lock manager error');
  3         END;
```

```
;  417        1405  3        CACHE[VCA$V_FIDC_VALID] = 0;
;  418        1406  2        END;
;  419        1407  2
;  420        1408  1 END;                                   ! end of routine DELETE_HEADER



                                                              .EXTRN  PMS$GL_FIDHIT, PMS$GL_FIDMISS
                                                              .EXTRN  ALLOCATION_LOCK
                                                              .EXTRN  INIT_FID_CACHE, READ_BLOCK
                                                              .EXTRN  ZERO_ON_ERROR, BUG$_BADFID
                                                              .EXTRN  SYS$ENQW, BUG$_XQPERR

                                         0BFC 00000          .ENTRY  DELETE_FID, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 1234
                                                                      R11                                         ;
                 5E             08    C2 00002               SUBL2   #8, SP                                      ;
                          98    AA    9F 00005               PUSHAB  -104(BASE)                                  ; 1285
           0000G CF             00    FB 00008               CALLS   #0, ALLOCATION_LOCK                         ; 1301
                 50       00    BE    D0 0000D               MOVL    @0(SP), R0                                  ; 1311
                 54       58    A0    D0 00011               MOVL    88(R0), CACHE                               ;
                 52             64    D0 00015               MOVL    (CACHE), FID_CACHE                          ; 1312
                          04    AC    D5 00018               TSTL    FILENUM                                     ; 1314
                                1D    13 0001B               BEQL    2$                                          ;
                 07       0B    A4    E8 0001D               BLBS    11(CACHE), 1$                               ; 1317
                                54    DD 00021               PUSHL   CACHE                                       ; 1318
           0000G CF             01    FB 00023               CALLS   #1, INIT_FID_CACHE                          ;
                          04    AC    DD 00028 1$:           PUSHL   FILENUM                                     ; 1319
           0000V CF             01    FB 0002B               CALLS   #1, RETURN_FILE_NUM                         ;
                 07             50    E9 00030               BLBC    R0, 2$                                      ;
                     00000000G  00    D6 00033               INCL    PMS$GL_FIDHIT                               ; 1322
                                04       00039               RET                                                 ; 1321
                          04    AC    D5 0003A 2$:           TSTL    FILENUM                                     ; 1327
                                03    12 0003D               BNEQ    4$                                          ;
                              0087    31 0003F 3$:           BRW     11$                                         ;
                 F9       0B    A4    E9 00042 4$:           BLBC    11(CACHE), 3$                               ; 1328
                     00000000G  00    D6 00046               INCL    PMS$GL_FIDMISS                              ; 1331
                                58    D4 0004C               CLRL    BEST_COUNT                                  ; 1332
                 53             01    CE 0004E               MNEGL   #1, VBN                                     ; 1333
                 5B       02    A2    3C 00051               MOVZWL  2(FID_CACHE), R11                           ; 1334
                 56       24    A2    9E 00055               MOVAB   36(FID_CACHE), R6                           ; 1337
                                50    D4 00059               CLRL    J                                           ;
                                31    11 0005B               BRB     8$                                          ;
     51          FC A640        01    C3 0005D 5$:           SUBL3   #1, -4(R6)[J], R1                           ;
     59             51 00001000 8F    C7 00063               DIVL3   #4096, R1, BLOCK                            ;
                 53             59    D1 0006B               CMPL    BLOCK, VBN                                  ; 1338
                                05    13 0006E               BEQL    6$                                          ;
                 53             59    D0 00070               MOVL    BLOCK, VBN                                  ; 1341
                                57    D4 00073               CLRL    COUNT                                       ; 1342
                                57    D6 00075 6$:           INCL    COUNT                                       ; 1344
                 58             57    D1 00077               CMPL    COUNT, BEST_COUNT                           ; 1345
                                06    1B 0007A               BLEQU   7$                                          ;
                 58             57    D0 0007C               MOVL    COUNT, BEST_COUNT                           ; 1348
                 55             53    D0 0007F               MOVL    VBN, BEST_VBN                               ; 1349
                 51       02    A2    3C 00082 7$:           MOVZWL  2(FID_CACHE), R1                            ; 1351
                 51             02    C6 00086               DIVL2   #2, R1                                      ;
                 51             58    D1 00089               CMPL    BEST_COUNT, R1                              ;
                                04    1E 0008C               BGEQU   9$                                          ;
```

```
                                                           K 10
DELFIL                                                     16-Sep-1984 00:17:11    VAX-11 Bliss-32 V4.0-742                     Page  11
V04-000                                                    14-Sep-1984 12:30:16    DISK$VMSMASTER:[F11X.SRC]DELFIL.B32;1          (3)

                     CB            50                5B  F3 0008E 8$:      AOBLEQ  R11, J, 5$                                 : 1334
                                   50          00    BE  D0 00092 9$:      MOVL    @0(SP), R0                                 : 1359
      55     38      A0            08                00  ED 00096          CMPZV   #0, #8, 56(R0), BEST_VBN
                                                     04  1A 0009C          BGTRU   10$
                                                       FEFF 0009E          BUGW                                               : 1360
                                                      0000* 000A0          .WORD   <BUG$_BADFID!4>
                                                     03  DD 000A2 10$:     PUSHL   #3                                         : 1362
                                                     01  DD 000A4          PUSHL   #1
                                   50          08    BE  D0 000A6          MOVL    @8(SP), R0
                                               30 B045   9F 000AA          PUSHAB  @48(R0)[BEST_VBN]
                           0000G   CF                03  FB 000AE          CALLS   #3, READ_BLOCK
                                   57                50  D0 000B3          MOVL    R0, BUFFER
                                            00A0     8F  BB 000B6          PUSHR   #^M<R5,R7>                                 : 1363
                                                     58  DD 000BA          PUSHL   BEST_COUNT
                           0000V   CF                03  FB 000BC          CALLS   #3, REMOVE_FILE_NUM
                                                     57  DD 000C1          PUSHL   BUFFER                                     : 1364
                           0000G   CF                01  FB 000C3          CALLS   #1, WRITE_BLOCK
                                                         04 000C8          RET                                                : 1327
                                   6D       0000G    CF  9E 000C9 11$:     MOVAB   ZERO_ON_ERROR, (FP)                        : 1376
                                               02    A2  B5 000CE 12$:     TSTW    2(FID_CACHE)                               : 1377
                                                     4C  13 000D1          BEQL    15$
                     50      24    A2                01  C3 000D3          SUBL3   #1, 36(FID_CACHE), R0                      : 1380
                     53            50 00001000       8F  C7 000D8          DIVL3   #4096, R0, VBN
                                   50          00    BE  D0 000E0          MOVL    @0(SP), R0                                 : 1381
      53     38      A0            08                00  ED 000E4          CMPZV   #0, #8, 56(R0), VBN
                                                     04  1A 000EA          BGTRU   13$
                                                       FEFF 000EC          BUGW                                               : 1382
                                                      0000* 000EE          .WORD   <BUG$_BADFID!4>
                                                     03  DD 000F0 13$:     PUSHL   #3                                         : 1384
                                                     01  DD 000F2          PUSHL   #1
                                   50          08    BE  D0 000F4          MOVL    @8(SP), R0
                                               30 B043   9F 000F8          PUSHAB  @48(R0)[VBN]
                           0000G   CF                03  FB 000FC          CALLS   #3, READ_BLOCK
                                   57                50  D0 00101          MOVL    R0, BUFFER
                                                     14  13 00104          BEQL    14$                                        : 1385
                                            0088     8F  BB 00106          PUSHR   #^M<R3,R7>                                 : 1388
                                                     7E  D4 0010A          CLRL    -(SP)
                           0000V   CF                03  FB 0010C          CALLS   #3, REMOVE_FILE_NUM
                                                     57  DD 00111          PUSHL   BUFFER                                     : 1389
                           0000G   CF                01  FB 00113          CALLS   #1, WRITE_BLOCK
                                                     B4  11 00118          BRB     12$                                        : 1385
                                               02    A2  B4 0011A 14$:     CLRW    2(FID_CACHE)                               : 1392
                                                     AF  11 0011D          BRB     12$                                        : 1377
                                               04    A2  D5 0011F 15$:     TSTL    4(FID_CACHE)                               : 1394
                                                     25  13 00122          BEQL    16$
                              08   AE          04    A2  D0 00124          MOVL    4(FID_CACHE), LOCK_STATUS+4                : 1397
                                                     7E  7C 00129          CLRQ    -(SP)                                      : 1402
                                                     7E  7C 0012B          CLRQ    -(SP)
                                                     7E  7C 0012D          CLRQ    -(SP)
                                                     7E  D4 0012F          CLRL    -(SP)
                                   7E          4E    8F  9A 00131          MOVZBL  #78, -(SP)
                                               24    AE  9F 00135          PUSHAB  LOCK_STATUS
                                   7E                1E  7D 00138          MOVQ    #30, -(SP)
                       00000000G   00                0B  FB 0013B          CALLS   #11, SYS$ENQW
                                   04                50  E8 00142          BLBS    R0, 16$
                                                       FEFF 00145          BUGW                                               : 1403
                                                      0000* 00147          .WORD   <BUG$_XQPERR!4>
```

```
                        0B    A4          01  8A 00149 16$:      BICB2   #1, 11(CACHE)                                  ; 1405
                                              04 0014D           RET                                                    ; 1408

; Routine Size:  334 bytes,    Routine Base:  $CODE$ + 00D1
```

```
ROUTINE RETURN_FILE_NUM (FILE_NUMBER) : L_NORM =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!	This routine returns a file number to the volume's file number
!	cache. If the cache fills up as a result, it also sorts the
!	entries and returns failure status to signal the caller that the
!	cache should be emptied.
!
!
! CALLING SEQUENCE:
!	RETURN_FILE_NUM (ARG1)
!
! INPUT PARAMETERS:
!	ARG1: file number to return
!
! IMPLICIT INPUTS:
!	CURRENT_VCB: VCB of volume
!	CURRENT_UCB: UCB of volume
!
! OUTPUT PARAMETERS:
!	NONE
!
! IMPLICIT OUTPUTS:
!	NONE
!
! ROUTINE VALUE:
!	1 if success
!	0 if cache is now full
!
! SIDE EFFECTS:
!	file ID cache modified
!
!--

BEGIN

LOCAL
	CACHE		: REF BBLOCK,	! address of cache block
	FID_CACHE	: REF BBLOCK,	! address of file number cache
	J;				! cache index

BIND_COMMON;


! Scan the cache for an entry higher than the file number being returned.
! Shuffle the cache upward and insert the file number in order. If the
! cache fills up, return failure to cause a cache flush.
!

CACHE = .CURRENT_VCB[VCB$L_CACHE];
FID_CACHE = .CACHE[VCA$L_FIDCACHE];
J = 0;
UNTIL .J GEQU .FID_CACHE[VCA$W_FIDCOUNT]
DO
```

```
;  479          1466 3          BEGIN
;  480          1467 3          IF .VECTOR [FID_CACHE[VCA$L_FIDLIST], .J] GTRU .FILE_NUMBER
;  481          1468 3          THEN EXITLOOP;
;  482          1469 3          IF .VECTOR [FID_CACHE[VCA$L_FIDLIST], .J] EQL .FILE_NUMBER
;  483          1470 3          THEN RETURN 1;
;  484          1471 3          J = .J + 1;
;  485          1472 2          END;
;  486          1473 2
;  487          1474 2      CH$MOVE ((.FID_CACHE[VCA$W_FIDCOUNT]-.J)*4,
;  488          1475 2               VECTOR [FID_CACHE[VCA$L_FIDLIST], .J],
;  489          1476 2               VECTOR [FID_CACHE[VCA$L_FIDLIST], .J+1]);
;  490          1477 2      VECTOR [FID_CACHE[VCA$L_FIDLIST], .J] = .FILE_NUMBER;
;  491          1478 2      FID_CACHE[VCA$W_FIDCOUNT] = .FID_CACHE[VCA$W_FIDCOUNT] + 1;
;  492          1479 2
;  493          1480 2      .FID_CACHE[VCA$W_FIDCOUNT] LSSU .FID_CACHE[VCA$W_FIDSIZE]
;  494          1481 2      AND .CACHE[VCA$V_FIDC_VALID]
;  495          1482 2
;  496          1483 1      END;                                      ! end of routine RETURN_FILE_NUM



                                                 01FC 00000 RETURN_FILE_NUM:
                                                                   .WORD    Save R2,R3,R4,R5,R6,R7,R8                 ; 1409
                           50       98   AA   D0 00002             MOVL     -104(BASE), R0                            ; 1461
                           58       58   A0   D0 00006             MOVL     88(R0), CACHE                             ;
                           56            68   D0 0000A             MOVL     (CACHE), FID_CACHE                        ; 1462
                                         57   D4 0000D             CLRL     J                                         ; 1463
                           50       24   A6   9E 0000F             MOVAB    36(FID_CACHE), R0                         ; 1467
     57     02   A6        10            00   ED 00013 1$:         CMPZV    #0, #16, 2(FID_CACHE), J                  ; 1464
                                         11   1B 00019             BLEQU    3$                                        ;
                     04   AC           6047   D1 0001B             CMPL     (R0)[J], FILE_NUMBER                      ; 1467
                                         0A   1A 00020             BGTRU    3$                                        ;
                                         04   12 00022             BNEQ     2$                                        ; 1469
                           50            01   D0 00024             MOVL     #1, R0                                    ; 1470
                                              04 00027             RET                                                ;
                                         57   D6 00028 2$:         INCL     J                                         ; 1471
                                         E7   11 0002A             BRB      1$                                        ; 1464
                           51       02   A6   3C 0002C 3$:         MOVZWL   2(FID_CACHE), R1                          ; 1474
                           51            57   C2 00030             SUBL2    J, R1                                     ;
                           51            04   C4 00033             MULL2    #4, R1                                    ;
                                 04 A047      DF 00036             PUSHAL   4(R0)[J]                                  ; 1476
                                    6047      DF 0003A             PUSHAL   (R0)[J]                                   ;
                 9E        9E            51   28 0003D             MOVC3    R1, @(SP)+, @(SP)+                        ;
                     24 A647        04   AC   D0 00041             MOVL     FILE_NUMBER, 36(FID_CACHE)[J]             ; 1477
                                    02   A6   B6 00047             INCW     2(FID_CACHE)                              ; 1478
                                         50   D4 0004A             CLRL     R0                                        ; 1480
                           66       02   A6   B1 0004C             CMPW     2(FID_CACHE), (FID_CACHE)                 ;
                                         02   1E 00050             BGEQU    4$                                        ;
                                         50   D6 00052             INCL     R0                                        ;
     51     0B   A8        01            00   EF 00054 4$:         EXTZV    #0, #1, 11(CACHE), R1                     ; 1481
                           51            51   D2 0005A             MCOML    R1, R1                                    ;
                           50            51   CA 0005D             BICL2    R1, R0                                    ;
                                              04 00060             RET                                                ; 1483

; Routine Size:  97 bytes,    Routine Base:  $CODE$ + 021F
```

```
1 ROUTINE REMOVE_FILE_NUM (COUNT, VBN, BUFFER) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine removes the specified entries from the file ID cache
1 !       and marks then free in the index file bitmap block supplied.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       REMOVE_FILE_NUM (ARG1, ARG2, ARG3)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: number of entrues to remove (0 to remove all)
1 !       ARG2: VBN of bitmap buffer
1 !       ARG3: address of bitmap buffer
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: address of volume VCB
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       file ID cache altered, bitmap buffer modified
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       BUFFER          : REF BITVECTOR; ! bitmap buffer
2
2 LOCAL
2       FID_CACHE       : REF BBLOCK,   ! address of file number cache
2       K,                              ! counter of entries removed
2       J,                              ! index into cache
2       FILE_NUMBER,                    ! file number-1 of entry
2       BITPOS;                         ! bit position in buffer
2
2 BIND_COMMON;
2
2 ! Scan the file ID cache for entries whose bitmap VBN match those of the
2 ! buffer. When one is found, clear the corresponding bit in the bitmap,
2 ! decrement the count in the cache, and shuffle down the remaining entries
2 ! to keep the cache compacted.
2 !
2
2 FID_CACHE = .BBLOCK [.CURRENT_VCB[VCB$L_CACHE], VCA$L_FIDCACHE];
2 K = .COUNT;
```

```
;   555        1541  2 J = 1;
;   556        1542  2 DO
;   557        1543  3     BEGIN
;   558        1544  3     FILE_NUMBER = .VECTOR [FID_CACHE[VCA$L_FIDLIST], .J-1] - 1;
;   559        1545  3     IF .FILE_NUMBER / 4096 EQL .VBN
;   560        1546  3     THEN
;   561        1547  4         BEGIN
;   562        1548  4         BITPOS = .FILE_NUMBER<0,12>;
;   563        1549  4         BUFFER[.BITPOS] = 0;
;   564        1550  4         CH$MOVE ((.FID_CACHE[VCA$W_FIDCOUNT]-.J)*4,
;   565        1551  4                  VECTOR [FID_CACHE[VCA$L_FIDLIST], .J],
;   566        1552  4                  VECTOR [FID_CACHE[VCA$L_FIDLIST], .J-1]);
;   567        1553  4         FID_CACHE[VCA$W_FIDCOUNT] = .FID_CACHE[VCA$W_FIDCOUNT] - 1;
;   568        1554  4         J = .J - 1;
;   569        1555  4         K = .K - 1;
;   570        1556  3         END;
;   571        1557  3     J = .J + 1;
;   572        1558  3     END
;   573        1559  2 UNTIL .K EQL 0 OR .J GTRU .FID_CACHE[VCA$W_FIDCOUNT];
;   574        1560  2
;   575        1561  2 ! If we have freed file numbers in a block that precedes the current bitmap
;   576        1562  2 ! scan point, reset the scan point.
;   577        1563  2 !
;   578        1564  2
;   579        1565  2 IF .VBN LSSU .CURRENT_VCB[VCB$B_IBMAPVBN]
;   580        1566  2 THEN CURRENT_VCB[VCB$B_IBMAPVBN] = .VBN;
;   581        1567  2
;   582        1568  2 1
;   583        1569  1 END;                                      ! end of routine RETURN_FILE_NUM



                                        0BFC 00000 REMOVE_FILE_NUM:
                                                           .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R11              ; 1484
                 50        98    AA  D0 00002              MOVL     -104(BASE), R0                                ; 1539
                 56        58    B0  D0 00006              MOVL     @88(R0), FID_CACHE                            ;
                 5B        04    AC  D0 0000A              MOVL     COUNT, K                                      ; 1540
                 58              01  D0 0000E              MOVL     #1, J                                         ; 1541
       57      20 A648           01  C3 00011  1$:         SUBL3    #1, 32(FID_CACHE)[J], FILE_NUMBER             ; 1544
       50        57 00001000     8F  C7 00017              DIVL3    #4096, FILE_NUMBER, R0                        ; 1545
            08   AC              50  D1 0001F              CMPL     R0, VBN                                       ;
                                 27  12 00023              BNEQ     3$                                            ;
  59   57        0C              00  EF 00025              EXTZV    #0, #12, FILE_NUMBER, BITPOS                  ; 1548
       00   0C   BC              59  E5 0002A              BBCC     BITPOS, @BUFFER, 2$                           ; 1549
                 50        02    A6  3C 0002F  2$:         MOVZWL   2(FID_CACHE), R0                              ; 1550
                 50              58  C2 00033              SUBL2    J, R0                                         ;
                 50              04  C4 00036              MULL2    #4, R0                                        ;
                            20 A648  DF 00039              PUSHAL   32(FID_CACHE)[J]                              ; 1552
                            24 A648  DF 0003D              PUSHAL   36(FID_CACHE)[J]                              ;
       9E        9E              50  28 00041              MOVC3    R0, @(SP)+, @(SP)+                            ;
                           02    A6  B7 00045              DECW     2(FID_CACHE)                                  ; 1553
                                 58  D7 00048              DECL     J                                             ; 1554
                                 5B  D7 0004A              DECL     K                                             ; 1555
                                 58  D6 0004C  3$:         INCL     J                                             ; 1557
                                 5B  D5 0004E              TSTL     K                                             ; 1559
```

```
                                        08  13 00050         BEQL    4$                                  ;
          58      02  A6        10      00  ED 00052         CMPZV   #0, #16, 2(FID_CACHE), J            ;
                                        B7  1E 00058         BGEQU   1$                                  ;
                                50  98  AA  D0 0005A 4$:     MOVL    -104(BASE), R0                      ; 1565
    08   AC      3A  A0         08      00  ED 0005E         CMPZV   #0, #8, 58(R0), VBN                 ;
                                        05  1B 00065         BLEQU   5$                                  ;
                         3A  A0     08  AC  90 00067         MOVB    VBN, 58(R0)                         ; 1566
                             50         01  D0 0006C 5$:     MOVL    #1, R0                              ; 1569
                                            04 0006F         RET                                         ;

; Routine Size:  112 bytes,    Routine Base:  $CODE$ + 0280


;  584        1570  1
;  585        1571  1 END
;  586        1572  0 ELUDOM



;                               PSECT SUMMARY
;
;       Name                     Bytes                    Attributes
;
;   $CODE$                            752  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                         Library Statistics
;
;                                            -------- Symbols --------       Pages      Processing
;       File                                 Total    Loaded   Percent       Mapped     Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1             18619       50         0        1000         00:01.8



;                               COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DELFIL/OBJ=OBJ$:DELFIL MSRC$:DELFIL/UPDATE=(ENH$:DELFIL)

; Size:          752 code + 0 data bytes
; Run Time:         00:49.2
; Elapsed Time:     01:43.4
; Lines/CPU Min:     1915
; Lexemes/CPU-Min: 58358
; Memory Used:  262 pages
; Compilation Complete
```

0169 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DEACCS
LIS

DELETE
LIS

DIRSCN
LIS

DISPAT
LIS

CREHDR
LIS

DIRACC
LIS

DELBAD
LIS

CREFCB
LIS

CREWIN
LIS

DISPATCH
LIS

ENTER
LIS

DELFIL
LIS